Információorientált dokumentumosztályozás a magyar Wikipédián

Subecz Zoltán[1], Farkas Richárd[2]

[1] Szolnoki Főiskola
5000 Szolnok, Tiszaligeti sétány 14.
subecz@szolf.hu
[2] Szegedi Tudományegyetem, Informatikai tanszékcsoport
Szeged, Árpád tér 2.
rfarkas@inf.u-szeged.hu

Kivonat: Az *információorientált dokumentumosztályozás* egy olyan speciális többcímkés dokumentumosztályozási feladat, ahol az osztályozás nem a dokumentum egészének témája, hanem a dokumentumban található speciális információ alapján történik. Az ilyen típusú feladatokat általában úgy oldják meg, hogy ún. indikátorkifejezéseket keresnek a szövegben, majd analizálják azok szövegkörnyezetét, hogy kiszűrjék a hamis pozitív találatokat [1]. Ebben a munkában használunk egy módszert a *lokális tartalommódosítók* gépi tanulására. A módszer csak dokumentumszintű tanító címkéket használ fel. A tartalommódosítókat általánosan kezeljük, egy adott nyelvi jelenség detektálása helyett (pl. tagadás). Egy rendszerbe integráljuk a dokumentumosztályozást és a tartalommódosulás felismerését. Munkánkban magyar nyelvű Wikipédia-szócikkeket dolgoztunk fel ezzel a módszerrel. Az angol nyelvű szövegekhez használt nyelvi elemzőket helyettesítettük magyar nyelvre kidolgozott elemzőkkel. A cikkben vizsgált fő kutatási kérdés az, hogy a magyar nyelvi elemzők mennyiben járulnak hozzá a feladat megoldásához.

1 Bevezető

Munkánkban olyan többcímkés szövegosztályozási feladatot oldottunk meg, ahol a célosztályok a szövegből kinyerhető speciális információval állnak kapcsolatban és nem a dokumentum általános témájával. Ebben a feladatban a célinformáció a dokumentum egy egyedének, például egy személynek valamilyen tulajdonsága, de a feladat a dokumentum szintjén való osztályba sorolás. Ehhez hasonló feladat például a következő:

Páciensek dohányzási szokása egy gyakori megjegyzés a klinikai kórlapok szöveges részében [2]. Ebben az esetben a feladat speciális információ megtalálása a szövegben, pl. az adott páciens dohányzik, dohányzott a múltban, vagy egyáltalán nem dohányzott, de végül az alkalmazásnak a dokumentumot (páciens) kell osztályba sorolnia.

Munkánkban labdarúgókkal kapcsolatos Wikipédia-szócikkeket dolgoztunk fel. Ezekben a cikkekben a labdarúgókról – sok más egyéb mellett – leírják, hogy melyik

csapatban játszottak eddig. Alkalmazásunknak a cikkeket ezek alapján kell osztályokba sorolni.

Ezeknél a feladatoknál a célinformáció csak megemlítésre kerül a dokumentumban, a dokumentum nagy része az információkinyerési feladat szempontjából nem lényeges. Ezzel ellentétben az általános szövegosztályozási feladatoknál a cél a dokumentumok teljes tartalma alapján történő osztályozás. Ez nem egy megszokott információkinyerési feladat, mivel a cél a dokumentumok osztályozása, és a tanító adathalmaz is csak ezen a szinten áll rendelkezésre. Tehát ez a speciális feladat az információkinyerés és a dokumentumosztályozás között helyezkedik el. Speciális megközelítést igényel a megoldása és *információorientált dokumentumosztályozásnak* nevezzük a későbbiekben.

Korábbi munkák [2,3,4] bemutatták, hogy az információorientált dokumentumosztályozás hatékonyan megvalósítható ún. *indikátorkifejezések* kézi vagy statisztikai összegyűjtésével. Azonban ezek a munkák megmutatták az indikátorkifejezések lokális *szövegkörnyezetének* vizsgálatának fontosságát. Például a dohányzási szokás felismeréséhez néhány indikátor szó (pl. *dohányzik*, *cigaretta*) elégséges, viszont ezek környezetét meg kell vizsgálni ahhoz, hogy megállapítsuk, hogy a szerepük megváltozott-e (pl. hogy tagadva, vagy múlt időben vannak-e). Például:

A páciens azt mondta, hogy nem dohányzik.
A páciens 5 éve még dohányzott

A cikkben vizsgált fő kutatási kérdés az, hogy a magyar nyelvi elemzők mennyiben járulnak hozzá a feladat megoldásához. Először az angol szövegekre készült alkalmazást futtattuk magyar szövegeken is. Majd megnéztük, hogy a magyar nyelvű előfeldolgozási rész mennyiben javította a módszer helyességét. Általánosságban azt kerestük, hogy milyen nyelvi sajátosságokra kell megoldást találnunk a magyar nyelvű szövegek feldolgozásához. Például a magyar nyelv szabad szórendje miatt azt feltételeztük, hogy a függőségi elemzés kiaknázása fontosabb a magyar nyelvű szövegeknél, mint angolban.

Munkánkban a következő tapasztalatokra jutottunk: Az angol nyelvre kidolgozott programot változtatás nélkül alkalmazva a magyar szövegen a várakozásoknak megfelelően gyenge eredményeket kaptunk. Azonban magyar nyelvi elemzők (lemmatizáló, szófaji egyértelműsítő, függőségi elemző) alkalmazásával az eredmények jelentősen javultak, és megközelítették az eredetileg angol szövegekre kapott értékeket.

2 Tartalommódosító detektálás

Munkánkban egy egyszerű, de hatékony módszert alkalmazunk az információorientált dokumentumosztályozási feladat megoldásához [1], ami lehetővé teszi az indikátorkifejezések jelentésének megváltozásának észlelését. Ezeket *tartalommódosítóknak* nevezzük, és ezek azonosításának a feladatát pedig *tartalommódosító detektálásnak* (Content Shift Detection: CSD).

A rendszer bemenete egy dokumentumszinten címkézett tanító korpusz. Módszerünk kiválogatja az indikátorkifejezéseket és tanítja a CSD-t párhuzamosan. A helyi tartalommódosítókra fókuszálunk, és csak az indikátorkifejezést tartalmazó mondatokra koncentrálunk. Alapfeltevésünk az, hogy a CSD-t tudjuk az által tanítani, hogy megkeressük a tanító adathalmazban az indikátorkifejezések hamis pozitív (FP) előfordulásait.

A feldolgozott korpusz a Wikipédián szereplő labdarúgó-játékosok voltak. Ez egy információorientált dokumentumcímkézési feladat, mert a sportolóhoz tartozó cikkben csak röviden van megemlítve, hogy melyik csapatoknál játszott. A feladat többcímkés dokumentumosztályozás, mert egy labdarúgóhoz általában több csapat is tartozik.

Példák a magyar korpuszból az indikátorkifejezések tartalommódosulására: Mindkét példában a labdarúgó csak a Videoton FC csapatban játszott.

1. példa:
Bátyja Szakály Péter, a Debreceni VSC játékosa.

Ha csak a *Debreceni VSC* indikátorkifejezést nézzük, akkor azt gondolhatnánk, hogy a labdarúgó a Debreceni VSC játékosa. De a szövegkörnyezetből látszik, hogy nem ő, hanem a bátyja játszik abban a csapatban.

2. példa:
Bemutatkozó mérkőzése hazai pályán az MTK ellen volt.

Az MTK indikátorkifejezés arra utalna, hogy a játékos az MTK-ban játszik. De a szövegkörnyezetet megvizsgálva látszik, hogy az MTK csapat ellen játszottak.

Példa indikátorkifejezés kiválasztásra: A Győri ETO FC csapatához a következő indikátorkifejezéseket választotta ki az alkalmazás: [győr, a rába, a győri].

A Wikipédia-kategóriákon taníthatjuk az osztályozót ismeretlen szövegek címkézésére. Az így kapott modellel például adhatunk olyan csapatneveket is egy-egy szócikkhez, amelyik nincs feltüntetve, azaz automatikusan javíthatjuk a Wikipédia kategória-hozzárendelését.

Általában az információorientált dokumentumosztályozásnál a tanító példák rendelkezésre állnak (pl. Wikipédia-kategóriák), így nem kell kézzel annotálni a szövegeket. Ez egy nagy előnye ennek a módszernek.

Ha a dokumentumcímkék rendelkezésre állnak tanítási időben, akkor egy iteratív módszert használhatunk a CSD és az indikátorszelekció együttes tanítására. A tanítási fázisnak két kimenete van: az indikátorkifejezések halmaza és a CSD. A CSD egy bináris függvény, amely meghatározza, hogy egy indikátorkifejezés jelentése egy adott szövegkörnyezetben módosult-e. Azok a jó indikátorkifejezések, amelyek utalnak a hozzájuk tartozó osztálycímkére. Az iteráció minden lépésénél minden címkéhez kiválasztjuk az indikátorkifejezéseket a dokumentumhalmaz aktuális állapota alapján. Az indikátorkifejezés környezete segítségével tanítjuk a CSD-t, a hamis pozitív (FP) indikátor találatok a pozitív (módosított jelentés), míg a valódi pozitív (TP) találatai (nem módosított jelentés) a negatív példák a CSD számára. A tanult CSD-t alkalmazzuk a kiinduló adathalmazra, és töröljük a kiinduló dokumentumokból azokat a szövegrészeket, amelyeket a CSD módosultnak jelölt. Minél jobbak az indikátoraink,

annál jobban lehet tanítani a CSD-t. Egy ilyen tisztított dokumentumhalmazt használva jobb indikátorokat tudunk kiválasztani. Az iterációs lépéseket egy adott konvergenciakritériumig végezhetjük. A munkánkban három iterációt alkalmaztunk, mert a korábbi kísérletek azt mutatták, hogy későbbi iterációk már nem javítanak szignifikánsan az eredményeken [1].

3 A korpusz bemutatása

A feldolgozott korpusz a Wikipédián szereplő labdarúgó-játékosok voltak. Minden Wikipédia-szócikk végén megtalálható, hogy az adott cikk milyen kategóriákhoz tartozik.

Például Nyilasi Tiborhoz a következő kategóriák vannak rendelve: *Magyar labdarúgók, Labdarúgó-középpályások, A Ferencváros labdarúgói, Az FK Austria Wien labdarúgói, Magyar bajnoki gólkirályok, Magyar labdarúgóedzők, Az FTC vezetőedzői, Az év magyar labdarúgói, Az 1978-as világbajnokság labdarúgói, Az 1982-es világbajnokság labdarúgói, Várpalotaiak, 1955-ben született személyek.*

Egy ilyen kategória a **Magyar labdarúgók** is.

A Wikipédia aktuális állapotát rendszeresen lementik XML formátumú ún. DUMP fájlba (2 GB).[1] Ezen fájl letöltése után kiválogattuk azokat a szócikkeket, amelyekhez a Magyar labdarúgók kategória is volt rendelve (2069 szócikk).

Először elvégeztük a Wikipédia-szövegek tisztítását, eltávolítottuk a feladathoz nem tartozó részeket a szövegekből. Ennek és még további tisztítási lépéseknek a segítségével az XML-dokumentumból elkészítettünk egy sima text formátumú szöveget, amely már csak a szócikkek szöveges részét tartalmazza.

Készítettünk egy szövegfájlt, amelybe kigyűjtöttük minden játékoshoz, hogy mely csapatokban játszott. Ezek alapján kigyűjtöttük, hogy melyik csapathoz hány játékos tartozik, és kiválasztottuk a legismertebb klubokat: az első tíz csapatot, amelyikhez a legtöbben tartoznak. (1. táblázat, 2. oszlop)

A játékosoknál a csapatnevekre nem egységes volt a hivatkozás a kategóriáknál sem. Például a Vasas SC névvel hivatkoznak a Budapesti Vasas SC csapatra, vagy DVTK névvel a Diósgyőri VTK csapatra. Minden csapatra megtalálható a Wikipédián, hogy milyen neveken szerepelt a múltban. Ezek alapján egységesítettük ennek a tíz csapatnak a múltbeli hivatkozásait. Így kaptuk a 1. táblázat 3. oszlopában látható előfordulásokat.

A korpuszt véletlenszerűen tanító és kiértékelő részekre bontottuk: 300 dokumentumot tettünk a kiértékelő részbe, a maradékot pedig a tanító részbe.

Korpuszhibákat az osztályozás első eredményeinek elemzése közben is észrevettünk. A nem releváns (FP) és a nem szereplő releváns (FN) osztályozási eredményeket összevetve a korpusz adataival azt tapasztaltuk, hogy a magyar Wikipédia szövegeinél sok helyen nincs jól megadva, hogy egy adott játékos melyik csapatoknál játszott. Volt olyan szócikk, ahol a szövegben szerepelt, hogy melyik csapatban játszott, de nem szerepelt a címkénél. És volt olyan, hogy a címkénél szerepelt, de a szövegben nem.

[1] http://meta.wikimedia.org/wiki/Data_dumps

Ez jelentősen befolyásolta a kiértékelés megbízhatóságát, ezért a korpuszt manuálisan végignéztük és javítottuk a címkéket a nem megfelelő helyeken.

1. táblázat: Az első 10 csapat kiválasztása.

Csapat neve	Az eredeti előfordulások	Az egységesített előfordulások	További korpuszjavítás
Ferencvárosi TC	425	433	363
MTK Budapest FC	314	335	260
Újpest FC	319	330	249
Budapest Honvéd FC	250	269	185
Budapesti Vasas SC	201	252	182
Győri ETO FC	201	203	139
Videoton FC	150	153	108
Debreceni VSC	136	142	110
Diósgyőri VTK	129	135	97
Szombathelyi Haladás	105	108	84
Összesen	**2230**	**2360**	**1777**

581 címkét kellett javítani a teljes korpuszon. Ez a javítás a korpusz csökkenését is eredményezte. Így 1015 tanító dokumentum és 255 kiértékelő dokumentum maradt, azaz átlagosan 1,4 címke tartozik egy dokumentumhoz (1. táblázat, 4. oszlop).

4 Magyar elemző modulok

Az angol szövegeket feldolgozó alkalmazáshoz saját modult illesztettünk, amely elvégzi a mondatokra és szavakra bontást és választhatóan a lemmatizálást is. Ehhez a magyarlanc programcsomagot használtuk fel [6]. A magyarlanc programcsomag[2] magyar nyelvű szövegek alap, nyelvi elemzésére szolgál. A csomag tisztán JAVA nyelvű modulokat tartalmaz, ami biztosítja a platformfüggetlenséget és a nagyobb rendszerekbe (például webszerverek) történő integrálhatóságot. A csomag magában foglal egy magyar nyelvre adaptált mondat- és tokenszegmentálót, illetve egy szófaji elemzőt és egy függőségi elemzőt [6]. A szófaji elemző (lemmatizáló és POS-tagger) a Stanford POS-tagger[3] egy módosított változata, amely az ismeretlen szavakra a morfológiai elemző által adott lehetséges elemzéseket használja fel. Azon szóalakok esetén, amelyek nem szerepelnek a tanító adatbázisban, egy morfológiai elemző meghatározza a lehetséges elemzések halmazát, majd a szófaji egyértelműsítő modulnak ezen halmazból kell választania [5].

Az angol nyelvre készített program felhasználta a MorphAdorner csomag[4] mondatokra, szavakra bontó és lemmatizáló modulját, valamint a Stanford csomag

[2] http://www.inf.u-szeged.hu/rgai/magyarlanc
[3] http://nlp.stanford.edu/software/tagger.shtml
[4] http://morphadorner.northwestern.edu/

tokenizáló és PCFG parser[5] modulját. A magyar nyelvre készített alkalmazás ezeket a magyarlanc [6] programcsomaggal helyettesíti. Ez végzi el a mondatokra és szavakra bontást és a lemmatizálást is. A program függőségi elemző része az adott mondathoz meghatározza annak nyelvtani struktúráját, és ez alapján minden szóhoz meghatározza, hogy melyik szóhoz kapcsolódik alárendelve nyelvtanilag, és hogy milyen szerepet tölt be a kapcsolatban. Az elemzési fa csomópontjaiban a szavak állnak, az ágai pedig a közöttük lévő kapcsolatok. Az elemzőfa kiinduló pontja (Root) mindig egy ige. Az alkalmazásba a Bohnet-parser függőségi elemzőt integrálták be.

Egy további javítást végeztünk el a magyar szövegek mondatokra bontó részén: Azokat a mondatokat, amelyek számmal kezdődtek, nem választotta szét az előző mondattól a tokenizáló. Mivel a korpuszon sok mondat kezdődik évszámmal, így ezeknél még külön két mondatra bontottuk azokat. Eddig a mondatok száma 10614 volt, ezzel a javítással 14741 lett. Látszik, hogy ez sok mondatot érintett. Ezen kívül voltak olyan mondatok, ahol az első betű közvetlenül az előző mondatot lezáró pont után következett szóköz nélkül. Ezeken a helyeken be kellett szúrni egy szóközt, hogy két külön mondatra válassza azt a tokenizáló.

5 Az indikátorkifejezések (jellemzők) kigyűjtése

Az indikátorkifejezések szavak sorozata, amelyek jelenléte utal a pozitív osztályra. 1, 2 vagy 3 hosszúságú kifejezéseket választottunk ki. Az indikátorkifejezések kiválasztására több fajta algoritmus áll rendelkezésre. A munkánkban egy jellemzőkiértékelésen alapuló mohó algoritmust alkalmaztunk az indikátorkifejezések kiválasztására az összes kifejezés halmazából. Az indikátorkiválasztási célunk az volt, hogy az összes pozitív dokumentumot kiválasszuk, miközben minél kevesebb nem releváns esetet kapjunk. A mohó algoritmus iterációnként kiválasztja a legjobb kifejezést egy jellemzőkiválasztó metrika alapján [1].

6 Dokumentumosztályozás

A Wikipédián minden játékoshoz ki van gyűjtve, hogy milyen csapatokban játszott eddig. Ezt a kigyűjtést címkézésnek is tekinthetjük. Puskás Ferencnél a következő csapatok vannak feltüntetve: *Budapest Honvéd FC, Real Madrid CF*.

Mivel egy játékoshoz általában több klub is tartozik, ezért ez egy többcímkés osztályozási feladat. Vizsgálatunkban nem foglalkoztunk címkék közötti függőségekkel, így a többcímkés osztályozást bináris (pozitív vagy negatív) osztályozásra vezettük vissza. Így a többcímkés osztályozó modellünk a bináris osztályozók egy halmaza. Az osztályozó nálunk az indikátor-előfordulást vizsgálja. Ha találtunk egy nem módosított indikátort a szövegben, akkor az osztálycímkét a dokumentumhoz rendeljük.

[5] http://nlp.stanford.edu/software/lex-parser.shtml

Abból a feltevésből indultunk ki, hogy míg az indikátorkifejezések osztályfüggők, addig a tartalommódosítók tanulhatók osztálytól függetlenül is. Ez a megközelítés elég sok tanító adatot ad a tartalommódosulás detektálásának tanításához.

A CSD egy bináris osztályozó, amely az indikátort tartalmazó mondat alapján eldönti, hogy módosított-e az indikátorok tartalma. A bináris tanuló szózsák és szintaktikai kapcsolat alapú jellemzőkön alapul [1].

7 Eredmények

7.1 Kiértékelési metrikák

A különböző módszerek kiértékeléséhez az alábbi metrikákat használtuk:

- **Dokumentumosztályozás CSD nélkül:** Ebben az esetben csak az indikátorszavak alapján végezzük el az osztályozást. Azaz ha talál egy indikátort a dokumentumban, akkor egyből a dokumentumhoz rendeli a címkét. A kiértékelési metrikák az egyes címkékhez tartozó bináris osztályozási feladat pozitív osztályának pontosság, fedés és F-értékei. A végső értékek az egyes címkék mikroátlagai.
- **Dokumentumosztályozás CSD-vel**: Ugyanaz, mint az előző, a tanított CSD alkalmazásával. A dokumentumban található minden indikátor esetén megkérdezzük a CSD-t, hogy módosult-e az indikátor. Ha van olyan indikátor a dokumentumban, amely nem módosult, akkor a dokumentumhoz rendeli az adott címkét.
- **CSD önállóan**: Itt címkétől független magát a CSD-t értékeljük ki. Azt mérjük, hogy hányszor jelezte egy indikátor-előfordulást módosítottnak a CSD, amikor tényleg az volt. Tehát egy releváns találat (TP) az az indikátor-előfordulás, amelyre a CSD igent mond, és a szóban forgó címke nem szerepel a dokumentum címkéi között (azaz ez a környezet módosított). Metrikának a „módosult” osztály pontosság, fedés és F-értékeit használjuk.

7.1 Vizsgálat az eredeti programmal az angol korpuszon

Az **angol nyelvű szövegeken** az eredeti program lemmatizálással, bigramokkal és függőségi elemzés nélkül a 2. táblázat 1. sorában látható eredményeket éri el. Látjuk, hogy a CSD szignifikánsan javította a dokumentumosztályozás eredményét. Megjegyezzük, hogy a CSD három információorientált dokumentumosztályozási feladaton is szignifikánsan jobbnak bizonyult, mint a standard dokumentumosztályozási módszerek [1], ezért a magyar nyelvű kísérleteknél csak ezzel foglalkoztunk. Ez volt a kiinduló rendszerünk. Innen vizsgáltuk a magyar szövegekre kidolgozott programrészek használatának hatását.

2. táblázat: A tartalommódosulás-detektálás eredményei pontosság/fedés/F-érték formátumban.

	Dokumentumosztályozás CSD nélkül	Dokumentumosztályozás CSD-vel	CSD önállóan
az angol korpuszon	0.851/ 0.854/0.853	0.911/0.855/0.882	0.965/0.454/0.617
a magyar korpuszon	0.696/ 0.857/0.768	0.724/0.849/0.782	0.909/0.136/0.236
magyar tokenizáló és lemmatizáló	0.728/0.949/0.824	0.760/0.944/0.842	0.913/0.152/0.260
indikátort követő szavakkal	**0.728/0.949/0.824**	**0.792/0.944/0.861**	**0.937/0.316/0.473**
bigramokkal, lemmatizálás nélkül	0.727/0.842/0.780	0.783/0.844/0.812	0.847/0.297/0.440
unigramokkal lemmatizálással	0.700/0.972/0.814	0.773/0.954/0.854	0.896/0.320/0.472

7.2 Az eredeti program tesztelése a magyar szövegeken

Ezek után az angol nyelvű szövegre kidolgozott programot teszteltük a **magyar javított szövegen** (2. táblázat 2. sora). Itt **az eredeti programon nem változtattunk**, csak az angol nyelvű korpusz helyett a magyar nyelvűt használtuk. Az eredmények várható módon jóval alatta maradtak az angol szövegeken kapott értékeknek (pontosság, fedés, F-érték értékek jelentős csökkenése). Hiszen itt még az angol nyelvre kidolgozott tokenizáló és lemmatizáló programokat használtuk.

7.3 A program tesztelése a magyar tokenizálóval és lemmatizálóval

A programba beépítettük a magyarlanc tokenizálóját és lemmatizálóját. A következő tesztelést ebben a környezetben végeztük el. Először lemmatizálással és bigramokkal vizsgáltuk az alkalmazás működését. A 2. táblázat 3. során látjuk, hogy ez 6 százalékpontnyi javulást eredményez.

7.4 Az indikátorkifejezéseket követő szavak vizsgálata

Az angol nyelvű program a CSD jellemzőinek az indikátorkifejezések előtti szavakat gyűjtötte ki (szózsák modell). A magyar nyelvűnél a szövegeken azt láttuk, hogy az indikátorkifejezések utáni szavak is módosíthatják a szerepüket.

3. példa:
Első gólját 2000. április 1-jén a Győri ETO FC ellen szerezte.

Itt a csapatnév: Győri ETO FC, indikátorkifejezés: Győri ETO. Ha csak az indikátorkifejezés előtti szavakat vizsgáljuk a mondatban, akkor abból arra is következtethetünk, hogy a Győri ETO FC csapatban játszott. De ha kigyűjtjük az indikátorkifejezés utáni szavakat is, akkor abból egyértelműen kiderül, hogy ezen a mérkőzésen nem a Győri ETO FC csapatban játszott.

Ezért a **mondat többi szavát is kigyűjtöttük** az osztályozáshoz. A 2. táblázat 4. során látjuk, hogy ennek hatására a CSD fedése 15 százalékponttal javult, ami a teljes dokumentumcímkézési feladat 2 százalékpontnyi javulását vonta magával.

7.5 Indikátorkifejezések

Megvizsgáltuk az osztályozó működését **unigramokkal** és **lemmatizálás nélkül** is. Az eredmények a várakozásoknak megfelelően alatta maradtak a bigramokkal és lemmatizálással való tesztelésnek (2. táblázat 5. és 6. sora).

A további kísérleteket a legeredményesebb (**Bigramokkal, lemmatizálással**) konfiguráció alkalmazásával hajtottuk végre.

7.6 Szintaktikai környezet

Eddig a CSD tanításakor a mondatban az indikátorkifejezés előtti és utáni szavakat használtuk fel (szózsák). Megvizsgáltuk az indikátorkifejezés szintaktikai környezetét is. A szózsák jellemzők mellé beillesztettük a mondatokra alkalmazott függőségi elemzésből kinyerhető jellemzőket is.

3. táblázat: A szintaktikai környezet vizsgálata pontosság/fedés/F érték formátumban.

	Dokumentum-osztályozás CSD-vel	CSD önállóan
Indikátortól a Root-ig	0.806/0.944/0.869	0.942/0.355/0.515
Lemma vizsgálata	0.811/0.941/0.871	0.928/0.376/0.536
Alany az útvonalon	0.810/0.938/0.870	0.945/0.376/0.538

Először minden indikátorkifejezéshez megkerestük és kigyűjtöttük a hozzá tartozó részfa szavait: az indikátorszótól a Root-ig tartó útvonalon az adott szót és a szülő csomóponttal való kapcsolatát. Ez azért fontos, mert a Root a mondat egy kiemelt szava, és az indikátorszótól a Root-ig lévő szavak erősen meghatározzák az indikátorszó szerepét. Ezeket is betettük az osztályozóba a **jellemző**k közé. Például ilyen jel-

legű kifejezéseket: DEPrln#MODE, vagy: DEPgovrln#ATT#ellen. A 3. táblázat 1. sorát a 2. táblázat 4. sorával összehasonlítva látjuk, hogy ezen jellemzők 4 százalékpontnyit javítottak a CSD-n, ami majdnem 1 százalékpontnyi javulást eredményezett a dokumentumosztályozási feladaton.

Az előző módszeren még annyit változtattunk, hogy a Root-ig tartó útvonalon nem a szót, hanem annak lemmáját tettük be (3. táblázat 2. sor). **Ez adta a vizsgálatunk legjobb eredményét**.

Az indikátorszótól a Root-ig végigmenve, ha valamelyik csomóponthoz alany (SUBJ) kapcsolódik, akkor az alanyt és a kapcsolat típusát felveszi az osztályozási jellemzők közé. Ez azért lehet fontos, mert a szócikkhez tartozó játékos gyakran a mondat alanya, így ezen tulajdonság kigyűjtése meghatározza az indikátorszóhoz való viszonyát (3. táblázat 3. sor). A pontosság/fedés/F-érték értékeken látjuk, hogy eredmények nem javultak az eddigi legjobb eredményhez képest.

9 Diszkusszió

A 2. és 3. táblázatok adatai alapján megállapíthatjuk, hogy a tokenizáló (2,9%), a lemmatizáló (további 3%) és az indikátort követő szavak kiválasztása (további 1,9%) jelentősen javította az eredményeket. A függőségi elemző használata is javította az eredményeket (további 1%).

Azt is megállapíthatjuk, hogy a magyar elemzőeszközök használatával hasonló eredményeket értünk el az angol eredményekhez képest. Azaz kijelenthetjük, hogy a feladatra hasonlóan jól működő megoldás adható magyar nyelven, mint angolra.

A táblázatok minden sorában látható, hogy a CSD használata jelentősen javította a dokumentumosztályozás eredményeit. A fedésértékek kis csökkenése mellett a pontosságértékek jelentős javulását látjuk (hamis pozitív találatokat szűrhetünk ki a CSD-vel).

10 Példák a CSD jellemzőterére

4. példa:

Az MTK labdarúgója volt, ahol három bajnoki címet és egy magyar kupát nyert a csapattal.

Indikátorszó: MTK

Szózsák tulajdonságok: labdarúgó, csapat, ahol, van, magyar, és, bajnoki, cím, egy, nyer, kupa, három

Szintaxisalapú tulajdonságok: DEPrln#ATT, DEPgov#volt, DEPrln#PRED, DEPgovrln#PRED#volt, DEPgov#labdarúgója, DEPgovrln#ATT#labdarúgója,

Itt a DEP tulajdonság a függőségi elemzésre utal, az *ATT, PRED* tulajdonságok az adott szó grammatikai függőségét jelölik az elemzési fában felette lévő szóhoz képest. Erre utal az *rln* jelzés is: a kapcsolat típusa. A *DEPgov* adja meg a felette levő szót.

5. példa:

*Bemutatkozó mérkőzése hazai pályán az MTK **ellen** volt, ahol tizenegy percet játszott.*

Indikátorszó: MTK

Szózsák tulajdonságok: pálya, ellen, az, hazai, játszik, ahol, mérkőzés, van, perc, tizenegy, bemutatkozó

Szintaxisalapú tulajdonságok: DEPrln#ATT, DEPgov#ellen, DEPgovrln#ATT#ellen, DEPrln#MODE, DEPgov#volt, DEPgovrln#MODE#volt

Mindkét példánál egy naiv rendszer pozitív címkézést adna. Az első példánál ez igaz is, de a második példánál ez nem releváns találat (FP) lenne. A CSD tanításával azt várjuk, hogy az osztályozó a második mondatnál negatív jelzést adjon. Ennél a példánál a CSD megtanulta, hogy a szózsák tartalmazza az *ellen* szót, ami módosítja az MTK indikátorszó tartalmát.

11 Összegzés

Ebben a cikkben az információorientált dokumentumcímkézési feladatokkal foglalkoztunk és gépi tanulási módszereket vizsgáltunk meg a helyi tartalommódosulás detektálásához. Empirikusan bizonyítottuk, hogy a nem releváns indikátorkifejezések felismerhetőek CSD tanításával. A tanított CSD nem használ semmilyen feladat-, vagy doménspecifikus ismeretet, csak dokumentumszintű annotált címkéket. Egy rendszerbe integráltuk a dokumentumosztályozást és a tartalommódosulás felismerést.

Munkánkban magyar nyelvű Wikipédia-szócikkeket dolgoztunk fel ezzel a módszerrel. Kiválasztottuk a magyar labdarúgók szócikkeivel kapcsolatos korpuszt, amelyet manuálisan javítottunk. Az angol nyelvű szövegekhez használt nyelvi elemzőket helyettesítettük magyar nyelvre kidolgozott tokenizáló, lemmatizáló és függőségi elemző modulokkal. A tokenizáló (2,9%), a lemmatizáló (3%) és az indikátort követő szavak kiválasztása (1,9%) jelentősen javította az eredményeket. A függőségi elemző használata is javította az eredményeket (további 1%). A magyar nyelvi modulok összesen így 8,9 százalékponttal (közel 40 százalékos hibacsökkentés) javították a dokumentumcímkézés hatékonyságát és az angol feladaton elért eredményekhez hasonló eredményt értünk el a magyar korpuszon.

Köszönetnyilvánítás

Jelen kutatást a futurICT.hu nevű, TÁMOP-4.2.2.C-11/1/KONV-2012-0013 azonosítószámú projekt támogatta az Európai Unió és az Európai Szociális Alap társfinanszírozása mellett.

Hivatkozások

1. Farkas, R.: Learning Local Content Shift Detectors from Document-level Information. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. John McIntyre Conference Centre, Edinburgh, UK (2011) 759–770
2. Uzuner, Ö., Goldstein, I., Luo, Y., Kohane, I.: Identifying Patient Smoking Status from Medical Discharge Records. Journal of American Medical Informatics Association, Vol. 15, No. 1 (2008) 14–24
3. Uzuner, Ö.: Recognizing obesity and comorbidities in sparse data. Journal of American Medical Informatics Association, Vol. 16, No. 4 (2009) 561–70
4. Pestian, J. P., Brew, C., Matykiewicz, P., Hovermale, D.J., Johnson, N., Cohen, K. B., Duch, W.: A shared task involving multi-label classification of clinical free text. In: Proceedings of the ACL Workshop on BioNLP (2007) 97–104
5. Zsibrita J., Vincze V., Farkas R.: Ismeretlen kifejezések és a szófaji egyértelműsítés. In: Tanács A., Vincze V. (szerk.): VII. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2010) 275–283
6. Zsibrita J., Vincze V., Farkas R.: magyarlanc 2.0: szintaktikai elemzés és felgyorsított szófaji egyértelműsítés. In: Tanács A., Vincze V. (szerk.): IX. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Tudományegyetem, Szeged (2013) 368–374